PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-135248
(43)Date of publication of application : 10.05.2002

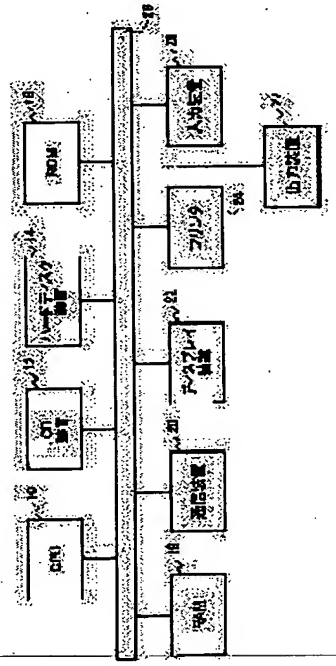
(51)Int.Cl. H04L 12/22
G06F 13/00
H04L 12/24
H04L 12/26

(21)Application number : 2000-320008
(22)Date of filing : 19.10.2000

(71)Applicant : MIZOGUCHI FUMIO
KUREO:KK
(72)Inventor : MIZOGUCHI FUMIO

(54) NETWORK-MONITORING METHOD, NETWORK-MONITORING SYSTEM AND STORAGE MEDIUM RECORDING ITS PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a network-monitoring method and a network-monitoring system which can monitor the state of application of a network, detect unauthorized access in a network or its apprehension, and inform abnormality to a system manager on real time, and storage medium recording its program.
SOLUTION: User' events to a network are collected as log data. From the collected log data, a profile showing the state of ordinary application of the network is formed for each user. Newly collected log data are compared with the profile for each user. By displaying compared results of the newly collected log data, the state of application of the network can be monitored from the compared results of the newly collected log data which are displayed. Thereby the unauthorized access in the network or its apprehension can be detected, and the abnormality can be informed on real time.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－135248
（P2002－135248A）
(43)公開日　平成14年５月10日(2002.5.10)

(51)Int.Cl.'		識別記号	ＦＩ		テーマコード'(参考)	
Ｈ０４Ｌ	12/22		Ｇ０６Ｆ	13/00	３５１Ｎ	５Ｂ０８９
Ｇ０６Ｆ	13/00	３５１	Ｈ０４Ｌ	11/26		５Ｋ０３０
Ｈ０４Ｌ	12/24			11/08		
	12/26					

審査請求　未請求　請求項の数７　ＯＬ　（全 13 頁）

(21)出願番号　特願2000－320008(P2000－320008)

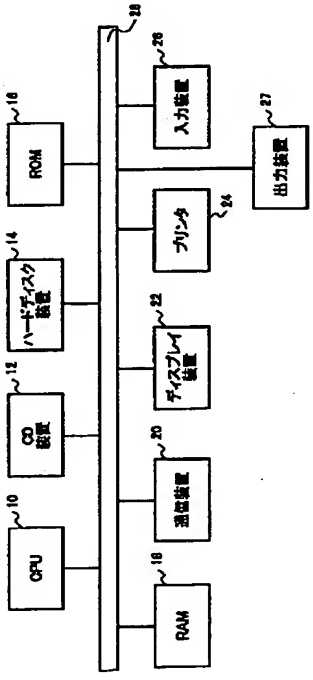
(22)出願日　平成12年10月19日(2000.10.19)

(71)出願人　597172890
溝口　文雄
東京都目黒区目黒１－17－３
(71)出願人　595155484
株式会社クレオ
東京都港区高輪３丁目19番22号
(72)発明者　溝口　文雄
東京都目黒区目黒１－17－３
(74)代理人　100070150
弁理士　伊東　忠彦

Ｆターム(参考)　5B089 GA21 GB02 JB16 KA17 LB14
5K030 GA15 HA05 JA10 KX30 MA04
MC09

(54)【発明の名称】　ネットワーク監視方法、ネットワーク監視システム及びそのプログラムを記録した記録媒体

(57)【要約】
【課題】　本発明は、ネットワーク使用状況を監視してネットワーク不正侵入またはそのおそれを検知し、リアルタイムにシステム管理者に異常を知らせることのできるネットワーク監視方法、ネットワーク監視システム及びそのプログラムを記録した記録媒体を提供することを目的とする。
【解決手段】　ネットワークに対するユーザのイベントをログデータとして収集し、収集されたログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成し、新たに収集したログデータをプロファイルとユーザ毎に照合し、新たに収集したログデータの照合結果を表示することにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視でき、ネットワーク不正侵入またはそのおそれを検知することができ、リアルタイムに異常を知らせることができる。



【特許請求の範囲】

【請求項１】　ネットワークに対するユーザのイベントをログデータとして収集し、
収集されたログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成し、
新たに収集したログデータを前記プロファイルとユーザ毎に照合し、
前記新たに収集したログデータの照合結果を表示することを特徴とするネットワーク監視方法。

【請求項２】　請求項１記載のネットワーク監視方法において、
前記新たに収集したログデータの照合結果に応じて前記ログデータの表示形態を異ならせて表示することを特徴とするネットワーク監視方法。

【請求項３】　ネットワークに対するユーザのイベントをログデータとして収集してデータベースに格納する収集手段と、
前記データベースに格納されているログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成するプロファイル生成手段と、
前記収集手段で新たに収集したログデータを前記プロファイルとユーザ毎に照合する照合手段と、
前記新たに収集したログデータの照合結果を表示する表示手段とを有することを特徴とするネットワーク監視システム。

【請求項４】　請求項３記載のネットワーク監視システムにおいて、
前記表示手段は、前記新たに収集したログデータの照合結果に応じて前記ログデータの表示形態を異ならせることを特徴とするネットワーク監視システム。

【請求項５】　請求項３または４記載のネットワーク監視システムにおいて、
前記表示手段は、収集したログデータを、ユーザ毎に、かつ、時間の推移に応じて２次元的に表示することを特徴とするネットワーク監視システム。

【請求項６】　請求項３乃至５のいずれか記載のネットワーク監視システムにおいて、
前記プロファイル生成手段は、パラメータ自動調整機能を持つＩＬＰシステムで構成されることを特徴とするネットワーク監視システム。

【請求項７】　コンピュータを、
ネットワークに対するユーザのイベントをログデータとして収集してデータベースに格納する収集手段と、
前記データベースに格納されているログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成するプロファイル生成手段と、
前記収集手段で新たに収集したログデータを前記プロファイルとユーザ毎に照合する照合手段と、
前記新たに収集したログデータの照合結果を表示する表示手段として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ネットワーク監視方法、ネットワーク監視システム及びそのプログラムを記録した記録媒体に関し、特に、ネットワークのアクセス状況を監視してネットワーク不正侵入等を検知するネットワーク監視方法、ネットワーク監視システム及びそのプログラムを記録した記録媒体に関する。

【０００２】

【従来の技術】従来より、ネットワークまたはネットワークの構成機器に対し、ネットワークの外部または内部から不正にアクセスするネットワーク不正侵入は社会的問題となっており、ネットワーク不正侵入を検知できる検知システムの登場が望まれている。上記のネットワーク不正侵入は年々増加しており、近年の調査によると、外部犯行と共に、内部犯行によるネットワーク不正侵入の被害件数も無視できないくらい多い。このため、侵入検知への期待が高まっている。

【０００３】従来より、統計や機械学習を利用してシステムの正常な運用状態を把握し、例外を検出することにより異常検知（ａｎｏｍａｌｙ　ｄｅｔｅｃｔｉｏｎ）を行うという研究がある。この研究は、電話の詐欺を検知するために普段の通話記録を機械学習によりルール化し、モニタリングするというものである。

【０００４】また、ユーザのコマンドログに対してベイジアンネットを利用して行動を分析する研究や、機械学習によりコマンドログのシーケンスを解析する研究などがある。

【０００５】

【発明が解決しようとする課題】しかし、既存の研究は未だ検討段階であり、ネットワークシステムへの不正アクセスが行われた段階で、ネットワーク不正侵入を検知してネットワーク管理者に異常を知らせるする具体的な技術はなかった。

【０００６】本発明は、上記の点に鑑みなされたもので、ネットワーク使用状況を監視してネットワーク不正侵入またはそのおそれを検知し、リアルタイムにシステム管理者に異常を知らせることのできるネットワーク監視方法、ネットワーク監視システム及びそのプログラムを記録した記録媒体を提供することを目的とする。

【０００７】

【課題を解決するための手段】請求項１に記載の発明は、ネットワークに対するユーザのイベントをログデータとして収集し、収集されたログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成し、新たに収集したログデータを前記プロファイルとユーザ毎に照合し、前記新たに収集したログデータの照合結果を表示することにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視

できネットワーク不正侵入またはそのおそれを検知することができ、リアルタイムに異常を知らせることができる。

【０００８】請求項２に記載の発明は、請求項１記載のネットワーク監視方法において、前記新たに収集したログデータの照合結果に応じて前記ログデータの表示形態を異ならせて表示することにより、ログデータの表示形態から視覚的にネットワーク使用状況をリアルタイムに監視できネットワーク不正侵入を検知することができる。

【０００９】請求項３に記載の発明は、ネットワークに対するユーザのイベントをログデータとして収集してデータベースに格納する収集手段と、前記データベースに格納されているログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成するプロファイル生成手段と、前記収集手段で新たに収集したログデータを前記プロファイルとユーザ毎に照合する照合手段と、前記新たに収集したログデータの照合結果を表示する表示手段とを有することにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視でき、ネットワーク不正侵入またはそのおそれを検知することができ、リアルタイムに異常を知らせることができる。

【００１０】請求項４に記載の発明は、請求項３記載のネットワーク監視システムにおいて、前記表示手段は、前記新たに収集したログデータの照合結果に応じて前記ログデータの表示形態を異ならせることにより、ログデータの表示形態から視覚的にネットワーク使用状況をリアルタイムに監視でき、ネットワーク不正侵入として検知されたログデータのチェックを容易に行うことができる。

【００１１】請求項５に記載の発明は、請求項３または４記載のネットワーク監視システムにおいて、前記表示手段は、収集したログデータを、ユーザ毎に、かつ、時間の推移に応じて２次元的に表示することにより、大量のログデータの全貌を表示することが可能となる。

【００１２】請求項６に記載の発明は、請求項３乃至５のいずれか記載のネットワーク監視システムにおいて、前記プロファイル生成手段は、パラメータ自動調整機能を持つＩＬＰシステムで構成されることにより、最良の性能パラメータ値を示すエラー率を用いて学習した結果の集合をプロファイルとして得ることができる。

【００１３】請求項７に記載の発明は、コンピュータを、ネットワークに対するユーザのイベントをログデータとして収集してデータベースに格納する収集手段と、前記データベースに格納されているログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成するプロファイル生成手段と、前記収集手段で新たに収集したログデータを前記プロファイルとユーザ毎に照合する照合手段と、前記新たに収集したログデータの照合結果を表示する表示手段として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体を用いることにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視でき、ネットワーク不正侵入またはそのおそれを検知することができ、リアルタイムに異常を知らせることができる。

【００１４】

【発明の実施の形態】図１は、本発明のネットワーク監視システムが構築されるコンピュータシステムの一実施例のブロック構成図を示す。同図中、ＣＰＵ１０はシステム全体の制御を司る。ＣＤ（コンパクトディスク）装置１２にセットされて再生されるＣＤには、ＣＰＵ１０で実行されるログチェッカー，プロファイリングエンジン，ビジュアルブラウザ等のプログラムが格納されている。ハードディスク装置１４にはブーツストラッププログラムやシステムプログラムが格納される他に、ログデータベースが格納される。また、ＲＯＭ１６には立ち上げ時にハードディスク装置１４からブートストラッププログラムやシステムプログラムを読み込むためのプログラムが格納されている。

【００１５】ＲＡＭ１８は、ＣＰＵ１０が実行するプログラムを一時記憶したり、演算結果を記憶するするための作業領域として使用される。通信装置２０はＬＡＮやインターネット等のネットワークに接続する。ディスプレイ装置２２はディスプレイ表示を行い、プリンタ２４は処理結果等をプリントアウトする。また、入力装置２６としてコマンド等を入力するキーボードと、表示画面におけるカーソル移動やクリックを行うためのマウス等が設けられ、出力装置２７として発音のためのスピーカが設けられている。上記の各装置間はシステムバス２８により接続されている。

【００１６】図２は、本発明のネットワーク監視システムのシステムアーキテクチャの一実施例のブロック図を示す。本発明システムは、ログチェッカー３０、ログデータベース３２、プロファイリングエンジン３４、及びビジュアルブラウザ３６から成る。ログチェッカー３０は、通信装置２０が接続されたネットワーク上を流れるユーザイベントを常時モニタリングして、そのユーザイベントをログデータとして収集しログデータベース３２に格納する。ログデータベース３２に収集されたログデータはプロファイリングエンジン３４、及びビジュアルブラウザ３６によって参照される。

【００１７】プロファイリングエンジン３４は、各ユーザの通常の振る舞いを表すプロファイルを生成する。このプロファイルの生成には、帰納論理プログラミング（ＩＬＰ：Ｉｎｄｕｃｔｉｖｅ　Ｌｏｇｉｃ　Ｐｒｏｇｒａｍｍｉｎｇ）を利用する。生成されたプロファイルは、ログチェッカー３０およびビジュアルブラウザ３６に送られる。

【００１８】ログチェッカー３０はこのプロファイルを利用し、ユーザのイベントをプロファイルと照合する。プロファイルに当てはまらないイベントは、そのユーザの通常の利用とは異なるため、不正なイベント即ちネットワーク不正侵入として検知され、検知情報がビジュアルブラウザ３６に通知される。

【００１９】ビジュアルブラウザ３６は、ログデータベース３２からログデータを取得し、ログデータそのものを表示すると共に、通常のイベントのログか、もしくは、不正なイベントのログかの分類を表示形態を変えることにより視覚的に表現する。また、システム管理者はビジュアルブラウザ３６を利用してイベントやプロファイルを編集することが可能である。これは、不正なイベントだけを検知するための完全なプロファイルの生成が困難であるため、システム管理者がイベントまたはプロファイルを編集するために必要となる。

【００２０】現実的には、不正なイベントは必ず検知するが、まれに正常なイベントを不正なイベントとして検知する程度のプロファイルをプロファイリングエンジン３４で生成する。この場合、システム管理者はネットワーク侵入等の不正なイベントとして検知されたイベントが本当に不正なものかどうかをチェックする必要があるが、ビジュアルブラウザ３６ではログデータを視覚的に表現するため、上記チェック作業が容易になる。ここで、対象とするユーザのイベントのログデータとしては、ユーザ名（ｗｈｏ）、マシン名（ｗｈｅｒｅ）、イベント名（ｗｈａｔ）、イベント引数（ｈｏｗ）、イベント実行時刻（ｗｈｅｎ）が上げられる。これは、どのユーザが、いつ、どこで、何を、どのように実行したかを表すものである。なお、ログデータとして、この５つの情報は必要不可欠ではなく、いずれかの情報を含んでいれば利用可能である。

【００２１】ビジュアルブラウザ３６は、ログデータの効率的な閲覧を可能とし、システム管理者に対して、視覚的にログのデータ構造を示すもので、通常のユーザの利用状況の把握や、ログチェッカー３０がネットワーク不正侵入を検知した場合に、どのようなイベントを実行したかの痕跡を発見し易くする。ビジュアルブラウザ３６は、ディスプレイ装置２２に、図３に示すログブラウザパネル、図４に示すプロパティパネル、図５に示すビジュアライズパネル、図６に示すプロファイルパネル、それぞれを表示する。

【００２２】図３は、ログブラウザパネルの一実施例を示す。同図中、ログ表示部４０に複数のログを行単位で表示する。また、テキストフィールド４２に、ログデータベース３２からログを取得する際の検索条件を記入する。この実施例では、ユーザ名「ｈｉｒａｉｓｈｉ」かつ、ホスト名「ｉｍｃｔ４５１」の検索条件を満足するログデータの取得することを指示している。テキストフィールド４２に記入された「&」は論理積を表すが、「&」の代わりに「|」を用いて論理和を指示することも可能である。

【００２３】Ｇｅｔボタン４３は、テキストフィールド４２に記入した条件によるログデータの取得を指示する。また、Ｃｌｅａｒボタン４４は、テキストフィールド４２のクリアを指示する。Ｂａｃｋボタン４５はＧｅｔボタン４３をクリックする一つ前に表示したログデータをログ表示部４０表示させることを指示する。ところで、テキストフィールド４２に未記入の状態でＧｅｔボタン４３をクリックすると、後述するプロパティパネルで設定された条件に相当するログデータが取得されてログ表示部４０に表示される。なお、テキストフィールド４２、Ｇｅｔボタン４３、Ｃｌｅａｒボタン４４、Ｂａｃｋボタン４５は、ログブラウザパネルだけでなくプロパティパネル、ビジュアライズパネル、プロファイルパネルにも共通に表示される。

【００２４】ログ表示部４０には、ログデータベース３２から取得した複数のログを行単位で表示される。各行左端の第１カラムにはユーザ名、第２カラムにはホスト名、第３カラムには実行したコマンド及びその引数、右端の第４カラムには実行した日付と時刻が表示される。また、ログ表示部４０に表示されているユーザ名やホスト名をマウスで選択してクリックすると、選択されたユーザ名やホスト名がテキストフィールド４２にコピーされ、自動的に項目（ユーザ名やホスト名）に関連するログデータがデータベース３２から取得される。このようにして、マウス操作によるログのブラウジングを可能としている。

【００２５】ところで、ログチェッカー３０によって、不正なイベントの通知があった場合には、ビジュアルブラウザ３６はカッコーアイコン４６が動かし、カッコーの鳴き声をスピーカより発音することによって、不正なイベントが発生したことをシステム管理者に通知する。また、不正なイベントに対応するログデータを例えば赤い文字で表示する。このように、視覚と聴覚によって、システム管理者に対し不正なイベント発生の通知が行われる。

【００２６】図４は、プロパティパネルの一実施例を示す。同図中、指定欄５０ではログデータを取得する期間（デフォルト値）を指定する。指定欄５２では、ログ表示部４０に表示されているユーザ名やホスト名等の項目をクリックしてテキストフィールド４２にコピーする際に、複数の項目の論理積と論理和のいずれを用いるかを設定する。指定欄５４ではログデータを取得するタイミングを指定する。また、指定欄５６ではログチェッカー３０との通信を行うためにログチェッカー３０のＩＰアドレス及びポート番号を指定する。

【００２７】同様に、プロファイリングエンジン３４のＩＰアドレス及びポート番号を指定するための図示されてない欄も設けられている。上記のログチェッカー３０

やプロファイリングエンジン３４のＩＰアドレス及びポート番号の指定は、ログチェッカー３０やプロファイリングエンジン３４がビジュアルブラウザ３６とは異なるコンピュータで動作する場合に必要となるものであって、図１に示すように、ログチェッカー３０やプロファイリングエンジン３４がビジュアルブラウザ３６と同一のＣＰＵで動作する場合には必要ない。

【００２８】図５は、ビジュアライズパネルの一実施例を示す。ビジュアライズパネルの最も重要な役割はマシンやネットワークの状態を容易に理解させることにある。図５において、ワイヤフレームによる楕円球６０は、縦軸にユーザ名及びマシン名からなる属性が表現され、横軸の左から右に時間の推移が表現された２次元構成である。縦横の線で区画された各セルには、ユーザが実行したコマンド（ログデータのコマンド）が表示される。

【００２９】楕円球６０の中央に向かうほどセルの表示領域は大きく、外枠に近くなるにつれてセルの表示領域は小さくなる。このため、外枠に近づくにつれて表示は小さくなるが、一度に大量のログデータの全貌を表示することが可能である。また、任意のセルをマウスでドラッグすることによって、楕円球６０を上下左右に回転するようにスクロールさせることによって、目的の属性（ユーザ及びマシン）及び時間の推移に沿ってログデータの内容を見ることが可能である。この時、楕円球６０の中央に位置するログデータの属性が左端部に選択ログ属性６２として表示され、また、楕円球６０の中央に位置するログデータのコマンドの実行時刻が下端部にコマンド実行時刻６４として表示される。

【００３０】また、新たなログデータが得られた場合には、最新のログデータを楕円球６０の中央のセル６０ａに表示するように、自動的にスクロールが行われる。図５に示す例は、楕円球６０は中央のセル６０ａに表示されているログデータ「ｌｓ」が得られスクロールされた状態を表している。

【００３１】更に、大量のログデータの中で、不正なイベントのログデータの発見を容易にするため、ログデータのコマンドに対して以下の4 つのカテゴリに分類を行い、カテゴリによって表示するセルの色分けを行う。

【００３２】第１のカテゴリは、Ｕｎ－ｒｕｌｅｄコマンドであり赤で表示する。このカテゴリは最も危険なカテゴリであり、当該属性のプロファイルに適合しないコマンド群である。つまり、このカテゴリに含まれるコマンドは、ユーザが本来実行しないコマンドであるため、そのユーザになりすましたネットワーク不正侵入者によって、実行されたコマンドである可能性が高い。

【００３３】第２のカテゴリは、Ｄａｎｇｅｒコマンドであり黄で表示する。このカテゴリには、ネットワークに関連するコマンドや、システム管理者用のコマンドが属し、例えばｓｕ，ｆｉｎｇｅｒ，ｆｔｐ，ｔｅｌｎｅｔ，ｒｌｏｇｉｎ，ａｌｉａｓ，ｃｏｎｆｉｇｕｒｅ，ｎｍａｐ等のコマンドである。これらのコマンドは、ネットワークへの不正侵入や、パスワードデータを盗み見るために利用されるものである。

【００３４】第３のカテゴリは、Ｓａｆｅｔｙコマンドであり青で表示する。このカテゴリは、当該属性のプロファイルに適合するコマンドや間違いなく安全なコマンドがこのカテゴリに含まれ、例えばｌｓ，ｃｄ，ｊｌａｔｅｘ，ｊａｖａ等のコマンドである。

【００３５】第４のカテゴリは、Ｕｎｋｎｏｗｎコマンドであり緑で表示する。このカテゴリは上記第１～第３のカテゴリに属さないコマンド群である。これは、多くの場合タイプミスされたコマンドや、ユーザが独自に作成したマクロのコマンドが含まれる。この第４のカテゴリも、ネットワーク不正侵入者によって作成されたコマンドの可能性は高い。

【００３６】大量のログデータを図５の楕円球６０に表示した場合、外枠に近いものの内容の表示は不可能であるが、色の表示は可能である。したがって、上記のようにカテゴリ毎にセルの色分けを行えば、色の表示をみるだけで不正なイベントを発見することが可能である。さらに、発見したセルをドラッグして楕円球６０の中央位置に移動させることでログデータの内容を読み取ることができ、何が実行されたのかを把握することが可能である。

【００３７】例えば、システム管理者は、重要度の高い順に、例えば、赤色、黄色、緑色のセルに注目し、その色のセルが存在する場合には、その内容のチェックを行う。これによって、システム管理者は大量なログデータを視覚的に理解し、その内容を確認することが可能となる。

【００３８】なお、カテゴリに応じて色を異ならせる以外に、文字のフォントを異ならせたり、点滅間隔を異ならせる等の、他の表示形態を異ならせる方法を採用しても良い。

【００３９】図６は、プロファイルパネルの一実施例を示す。同図中、プロファイル表示部７０には、プロファイリングエンジン３４によって生成されたプロファイルがユーザ毎にリストとして表示される。このリストはプロファイルを構成するルールを行毎に表示している。

【００４０】図６に示す例では、第１行の記述「－－－ｈｉｒａｉｓｈｉ－－－」がユーザ名を表している。リストの第２行の記述「＊，ｊａｖａ，」における「＊」は全てに当てはまることを示すワイルドカードであり、このユーザは「ｊａｖａ」というコマンドをマシンに関係なく利用するというルールである。以降の行にそのユーザの通常のコマンドのルールが続いて表示されている。第５行の記述は、このユーザはマシン名「ｉｍｃｔｅｒ０３」で、コマンド「ｋｔｅｒｍ」を実行し、そのとき、引数の一番目に「&」“を指定するというルール

である。第２行以降の各ルールは論理和の関係となる。

【００４１】プロファイリングエンジン３４によって生成されるプロファイルは完全ではないため、プロファイル表示部７０では、各行のルールを編集したり、新たなルールの追加やルールの削除が可能である。また、ボタン７２をクリックすることで全てのユーザについての新たなプロファイルをプロファイリングエンジン３４から取得する。また、欄７４によって、新たなプロファイルをプロファイリングエンジン３４から取得するタイミング（図示の例では６０秒毎に取得）を設定することができる。

【００４２】プロファイリングエンジン３４では帰納論理プログラミング（ＩＬＰ）システムを利用する。ＩＬＰシステムは、機械学習の一つの手法であり、一階述語論理に基づいた例からの関係学習、更に、逐次的な処理が可能なシステムである。そして、ＩＬＰは次のようなフレームワークに基づいている。正事例E^+、負事例E^-、それらを説明する背景知識Ｂを与えたとき、（１）式により正事例を含み、（２）式により負事例を排除する仮説Ｈを仮説探索により見つける。

【００４３】

【数１】

$$B \wedge H \models E^+ \quad \cdots\cdots(1)$$

$$B \wedge H \wedge E^- \not\models \phi \quad \cdots\cdots(2)$$

ここで、記号 $\models$ は伴意と呼び(1)式はB∧HからE⁺が導かれることを意味する。

ＩＬＰでは、ユーザが与えた宣言情報を利用して述語を結合する。これは、規則にのっとって関係データベース内の複数テーブルを結合するのと等しい。これは、従って、このテーブルに複数のログデータを格納し、そこから規則を導出する。本実施例では汎用性を考え、そのＩＬＰシステムをＪａｖａ言語によって実装する。

【００４４】ここでは、仮説生成を行なうとき、入出力モードと変数結合の深さを利用して、最も特殊な節をまず求める。そして、最も一般的な仮説から最も特殊な仮説（ボトム節）に向けて仮説を徐々に特殊化する。この時、仮説における正事例の包含率と仮説の長さを目的関数とし、含んでもよい負事例の割合（エラー率）を制約条件として与え、この目的関数が最大となる仮説を最良仮説、即ちルールとする。このように、ＩＬＰシステムは、学習時に負事例をどの程度含んでもよいかというエラー率を設定しなければならない。その設定次第でルールの性能が左右される。

【００４５】ここで、得られたルールが正事例と負事例をどの程度正しく説明できるかを示す精度（Ａｃｃｕｒａｃｙ）、正と判別する規則が正事例をどの程度判別できるかを示す感度（Ｓｅｎｓｉｔｉｖｉｔｙ）、負と判別する規則が負事例をどの程度判別できるかを示す特殊性（Ｓｐｅｃｉｆｉｃｉｔｙ）の３種類の性能パラメータを次のように定義する。

【００４６】正事例E^+が仮説Ｈで正として判断するＴＰ（Ｔｒｕｅ　Ｐｏｓｉｔｉｖｅ）と仮説Ｈで負として判断するＦＮ（Ｆａｌｓｅ　Ｎｅｇａｔｉｖｅ）とからなり、負事例E^-が仮説Ｈで正として判断するＦＰ（Ｆａｌｓｅ　Ｐｏｓｉｔｉｖｅ）と仮説Ｈで負として判断するＴＮ（Ｔｒｕｅ　Ｎｅｇａｔｉｖｅ）とからなるとき、性能パラメータは次の式で表される。

【００４７】感度＝ＴＰ／E^+

特殊性＝ＴＮ／E^-

精度＝（ＴＰ＋ＴＮ）／（E^+ ＋E^-）

これらの性能パラメータは、リサンプリング手法により求まる。そして、指定した性能パラメータを最大化し、指定されなかった残りの性能パラメータを制約条件として定義する。この時、性能パラメータの特徴として、エラー率を大きくすると感度はよくなり精度は悪くなる傾向がある。そこで、エラー率と各性能パラメータの関係を考慮に入れた探索により、最良の性能パラメータ値を示すエラー率を求める。その時のエラー率を用いて学習した結果の集合をプロファイルとする。

【００４８】図７は、プロファイリングエンジン３４内のＩＬＰシステムが実行するエラー率自動調整処理の一実施例のフローチャートを示す。

【００４９】同図中、ステップＳ１０でシステム管理者は最大化したい性能パラメータを決め、制約条件として残り２つの性能パラメータ値を与える。ステップＳ１２で停止条件を満足したか否かを判別して、停止条件を満足しない場合は、ステップＳ１４に進む。ここではクロスバリデーションやブートストラップのようなリサンプリング処理により訓練集合とテスト集合を作成する。

【００５０】次に、ステップＳ１６でＩＬＰシステムは訓練集合を学習し、それをテストして３つの性能パラメータを求める。そして、ステップＳ１８でシステム管理者が指定した制約条件を２つの性能パラメータが満足しているかどうかを確認する。ここで、満足している場合は、ステップＳ２０の調整プロセスに進んで、最大化したい性能パラメータが最大になるようにエラー率を更新してステップＳ１２に進みステップＳ１２～Ｓ１８を繰り返す。

【００５１】一方、ステップＳ１８で制約条件を満足している場合には、ステップＳ２２の最大化プロセスに進んで最大化したい性能パラメータが最大になるようにエラー率を更新してステップＳ１２に進みステップＳ１２～Ｓ１８を繰り返す。そして、最大化したい性能パラメータの値に変動がなくなると、ステップＳ１２で停止条件を満足してこの処理を終了する。

【００５２】上記の処理によって、ログデータベース３２のログイン状況とコマンドログを取得することにより、例えば次に示すようなルールが作成される。

【００５３】ｕｓｅｒ＿Ａ（Ｘ）：－ｌｏｇｉｎ

（X，’’ｉｍｃｔＸ’’，Ｙ），ｔｙｐｅ（Ｙ，’’ｌｐｒ、ｐｉｍｃ３，３’’）．
このルールは、ホスト名ｉｍｃｔＸではユーザＡはプリンタ利用のコマンドｌｐｒをタイプし、そのとき第３引数にネットワークプリンタを指定するｐｉｍｃ３を用いることを表している。このようなルールの集合をあるカテゴリのプロファイル集合とする。

【００５４】ログデータベース３２では、ファイルと関係データベースとを利用してログデータの管理を行う。関係データベースは例えば、「ＩＤ（識別番号）、ユーザ名、ホスト名、タイプされたコマンド（引数を含む）、タイプされたコマンド（引数を含まない）、タイプされた時刻」のテーブルと、「ＩＤ、タイプされたコマンド（引数を含む）、コマンド引数」のテーブルとの２つのテーブルから構成する。また、ファイルは例えば、上記関係データベースの各テーブルを１つのファイルとして、その中にＣＳＶ（Ｃｏｍｍａ　Ｓｅｐａｒａｔｅｄ　Ｖａｌｕｅ）形式で上記関係データベースと同一データを格納する。

【００５５】このように、ファイルと関係データベースを用いるのは、関係データベースが利用できない環境であっても、ファイルを利用できるようにしておくためであり、ファイルと関係データベースのいずれか一方だけを利用するものであっても良い。

【００５６】関係データベース、ファイル共に、関係データベースを演算するための言語であるＳＱＬ（Ｓｔｒａｃｔｕｒｅｄ　Ｑｕｅｒｙ　Ｌａｎｇｕａｇｅ）を用いて演算を行い、ログデータの加工と検索を行なう。図８に示すように、ビジュアルブラウザ３６のログブラウザパネル、プロパティパネル、ビジュアライズパネル、プロファイルパネルそれぞれは、必要な情報をログデータベース３２から取得するときに、演算要求「クエリ」をログデータベース３２に供給して、ログデータの加工及び検索を行い、その演算結果を得て、上記の各パネルに表示する。また、プロファイリングエンジン３４は、プロファイルを作成する際の関係抽出や仮説の包含数を導出するときに、そのための演算要求をログデータベース３２に供給してログデータベース３２を利用する。ログチェッカー３０は、分散型のログ収集を行なう。この場合、ログチェッカー３０は、図９に示すように、ログ収集サーバ８０と複数のモニタエージェント８２ａ～８２ｘとからなる。対象としたいログ（ユーザ名やホスト名）が決まれば、対象のログを監視するモニタエージェントを作成し、それにログを監視させる。ここでいう監視とは、ログの収集及びプロファイルとの照合による検知である。

【００５７】ログの収集では、各モニタエージェント８２ａ～８２ｘが収集したログをログ収集サーバ８０から例えばＴＣＰ／ＩＰ接続を利用してログデータベース３２の対象ファイル、または、対象関係データベースに付け加える。プロファイルとの照合による検知では、各モニタエージェント８２ａ～８２ｘは、プロファイリングエンジン３４で作成されたプロファイルと得られたログを照合し、そのログが異常の有無を示すフラグ等の検知情報をビジュアルブラウザ３６に供給する。

【００５８】図１０は、ログチェッカー３０が実行する処理の一実施例のフローチャートを示す。同図中、ステップＳ３０でネットワーク上を流れるログデータを取得する。次に、ステップＳ３２で取得したログデータをログデータベース３２に送信する。ステップＳ３４では取得したログデータをプロファイリングエンジン３４から供給されている対応するプロファイルと照合して、異常なログか否かを判別する。ここで、異常と判別されるとステップＳ３６で異常なログデータをビジュアルブラウザ３６に送信してステップＳ３０に進み、異常がないと判別されるとそのままステップＳ３０に進んで、上記の処理を繰り返す。

【００５９】図１１は、ログデータベース３２が実行する処理の一実施例のフローチャートを示す。同図中、ステップＳ４０でログチェッカー３０から送信されたログデータを受信し、ステップＳ４２でこのログデータを保存する。次に、ステップＳ４４でプロファイリングエンジン３４からログ取得要求があるか否かを判別する。ログ取得要求があった場合はステップＳ４６でログ取得要求があったログデータをプロファイリングエンジン３４に送信してステップＳ４８に進む。ログ取得要求がなかった場合はそのままステップＳ４８に進む。

【００６０】ステップＳ４８ではビジュアルブラウザ３６からログ取得要求があるか否かを判別する。ログ取得要求があった場合はステップＳ４９でログ取得要求があったログデータをビジュアルブラウザ３６に送信してステップＳ４０に進む。ログ取得要求がなかった場合はそのままステップＳ４０に進んで、上記の処理を繰り返す。

【００６１】図１２は、ビジュアルブラウザ３６が実行する処理の一実施例のフローチャートを示す。同図中、ステップＳ５０でログチェッカー３０またはログデータベース３２から送信されたログデータ、またはプロファイリングエンジン３４から送信されたプロファイルを受信する。次に、ステップＳ５２で受信したログデータは異常なログであるか否かを判別する。異常なログであればステップＳ５４で警報を鳴らし、ステップＳ５６に進む。異常なログでなければそのままステップＳ５６に進む。ステップＳ５６ではログデータの分類及び表示を行い、ステップＳ５０に進んで、上記の処理を繰り返す。

【００６２】図１３は、ステップＳ５６で実行されるログデータの分類及び表示処理の一実施例のフローチャートを示す。同図中、ステップＳ６０でログチェッカー３０から送信されたログデータのコマンドを、そのログデータのユーザ名に対応するプロファイルと照合して照合

できたか否かを判別する。照合できなかったログデータのコマンドについてはＵｎ－ｒｕｌｅｄコマンドであるので、ステップＳ６２に進んで赤で表示を行う。

【００６３】照合できたログデータのコマンドについてはステップＳ６４で、ｓｕ，ｆｉｎｇｅｒ，ｆｔｐ，ｔｅｌｎｅｔ，ｒｌｏｇｉｎ，ａｌｉａｓ，ｃｏｎｆｉｇｕｒｅ，ｎｍａｐ等のＤａｎｇｅｒコマンドと一致するか否かによりＤａｎｇｅｒコマンドに分類されるか否かを判別し、Ｄａｎｇｅｒコマンドに分類されると、ステップＳ６６に進んで黄で表示を行う。

【００６４】Ｄａｎｇｅｒコマンドに分類されないログデータのコマンドについてはステップＳ６８で、ｌｓ，ｃｄ，ｊｌａｔｅｘ，ｊａｖａ等のＳａｆｔｙコマンドと一致するか否かによりＳａｆｔｙコマンドに分類されるか否かを判別し、Ｓａｆｔｙコマンドに分類されると、ステップＳ７０に進んで青で表示を行う。Ｓａｆｔｙコマンドに分類されないログデータのコマンドについてはステップＳ７２でＵｎｋｎｏｗｎコマンドに分類して緑で表示を行う。

【００６５】図１４は、プロファイリングエンジン３４が実行する処理の一実施例のフローチャートを示す。同図中、ステップＳ８０でログデータベース３２に対し任意のユーザ名に対応する過去の全てのログデータの送信要求を行い、これによりログデータベース３２から送信されたログデータを受信する。次に、ステップＳ８２で受信した任意のユーザ名に対応する過去の全てのログデータからプロファイルを作成する。

【００６６】この後、ステップＳ８４で作成したプロファイルをログチェッカー３０に送信し、ステップＳ８６で作成したプロファイルをビジュアルブラウザ３６に送信したのち、ステップＳ８０に進んで、上記の処理を繰り返す。

【００６７】ここで、ユーザ名「ｙｏｓｈｉｉ」に対応して、図１５に示すログデータがログデータベース３２に格納されているものとする。図１５では、各２行で１つのコマンドを表す。第１行の「＋０９６４６１５６５８」はコマンドの実行時間であり、「ｉｍｃｔｕｔ０１」は実行マシン名であり、第２行の「ｊｌａｔｅｘ　ａ．ｔｅｘ」は実行コマンドである。この場合、プロファイリングエンジン３４は、図１５に示すログデータから図１６に示すプロファイルを生成する。

【００６８】図１６に示すユーザ名「ｙｏｓｈｉｉ」のプロファイルにおいて、第２行のプロファイル「＊，ｊｌａｔｅｘ，＊」は、図１５の第１，２行のログデータと第７，８行のログデータから生成され、両ログデータのマシン名，引数が異なるため、単にｊｌａｔｅｘというコマンドを利用する、といった意味のプロファイルである。図１６の第３行のプロファイルは、図１５の第３，４行のログデータと第９，１０行のログデータから生成されている。

【００６９】図１６の第４行のプロファイルは、図１５の第５，６行のログデータと第１１，１２行のログデータから生成され、両ログデータともマシン名はｉｍｃｔｕｔ０２で、１番目の引数に－Ｐｉｍｃ３が指定されているので、ｌｐｒというコマンドをｉｍｃｔｕｔ０２というマシンで引数の１番目に－Ｐｉｍｃ３を利用する、といった意味のプロファイルである。

【００７０】上記のプロファイルが得られた後、図１７に示すログデータが新たに得られた場合について説明する。図１７の第１，２行のログデータ及び第３，４行のログデータはプロファイル「＊，ｊｌａｔｅｘ，＊」に一致するためログチェッカー３０では正常と判別される。

【００７１】図１７の第５，６行のログデータはプロファイル「＊，ｄｖｉ２ｐｓ，＊」に一致するためログチェッカー３０では正常と判別される。図１７の第７，８行のログデータはプロファイル「ｉｍｃｔｕｔ０２，ｌｐｒ，－Ｐｉｍｃ３，１」に対し、１番目の引数－Ｐｉｍｃ３が異なるためログチェッカー３０では異常と判別され、この第７，８行のログデータはビジュアルブラウザ３６に送信される。

【００７２】

【発明の効果】上述の如く、請求項１に記載の発明は、ネットワークに対するユーザのイベントをログデータとして収集し、収集されたログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成し、新たに収集したログデータを前記プロファイルとユーザ毎に照合し、前記新たに収集したログデータの照合結果を表示することにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視でき、ネットワーク不正侵入またはそのおそれを検知することができ、リアルタイムに異常を知らせることができる。

【００７３】請求項２に記載の発明は、新たに収集したログデータの照合結果に応じて前記ログデータの表示形態を異ならせて表示することにより、ログデータの表示形態から視覚的にネットワーク使用状況をリアルタイムに監視できネットワーク不正侵入を検知することができる。

【００７４】請求項３に記載の発明は、ネットワークに対するユーザのイベントをログデータとして収集してデータベースに格納する収集手段と、前記データベースに格納されているログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成するプロファイル生成手段と、前記収集手段で新たに収集したログデータを前記プロファイルとユーザ毎に照合する照合手段と、前記新たに収集したログデータの照合結果を表示する表示手段とを有することにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視でき、ネットワーク不正侵入またはそのおそれを

検知することができ、リアルタイムに異常を知らせることができる。

【００７５】請求項４に記載の発明では、表示手段は、前記新たに収集したログデータの照合結果に応じて前記ログデータの表示形態を異ならせることにより、ログデータの表示形態から視覚的にネットワーク使用状況をリアルタイムに監視でき、ネットワーク不正侵入として検知されたログデータのチャックを容易に行うことができる。

【００７６】請求項５に記載の発明では、表示手段は、収集したログデータを、ユーザ毎に、かつ、時間の推移に応じて２次元的に表示することにより、大量のログデータの全貌を表示することが可能となる。

【００７７】請求項６に記載の発明では、プロファイル生成手段は、パラメータ自動調整機能を持つＩＬＰシステムで構成されることにより、最良の性能パラメータ値を示すエラー率を用いて学習した結果の集合をプロファイルとして得ることができる。

【００７８】請求項７に記載の発明は、コンピュータを、ネットワークに対するユーザのイベントをログデータとして収集してデータベースに格納する収集手段と、前記データベースに格納されているログデータからユーザ毎に通常のネットワーク使用状況を表すプロファイルを生成するプロファイル生成手段と、前記収集手段で新たに収集したログデータを前記プロファイルとユーザ毎に照合する照合手段と、前記新たに収集したログデータの照合結果を表示する表示手段として機能させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体を用いることにより、表示される新たに収集したログデータの照合結果からネットワーク使用状況を監視でき、ネットワーク不正侵入またはそのおそれを検知することができ、リアルタイムに異常を知らせることができる。

【図面の簡単な説明】

【図１】本発明のネットワーク監視システムが構築されるコンピュータシステムの一実施例のブロック構成図である。

【図２】本発明のネットワーク監視システムのシステムアーキテクチャの一実施例のブロック図である。

【図３】ログブラウザパネルの一実施例を示す図である。

【図４】プロパティパネルの一実施例を示す図である。

【図５】ビジュアライズパネルの一実施例を示す図である。

【図６】プロファイルパネルの一実施例を示す図である。

【図７】プロファイリングエンジン３４内のＩＬＰシステムが実行するエラー率自動調整処理の一実施例のフローチャートである。

【図８】ビジュアルブラウザ３６及びプロファイリングエンジン３４によるログデータベース３２の利用を説明するための図である。

【図９】分散型のログ収集を行なうログチェッカー３０の構成を説明するための図である。

【図１０】ログチェッカー３０が実行する処理の一実施例のフローチャートである。

【図１１】ログデータベース３２が実行する処理の一実施例のフローチャートである。

【図１２】ビジュアルブラウザ３６が実行する処理の一実施例のフローチャートである。

【図１３】ステップＳ５６で実行されるログデータの分類及び表示処理の一実施例のフローチャートである。

【図１４】プロファイリングエンジン３４が実行する処理の一実施例のフローチャートである。

【図１５】ログデータベース３２に格納されているログデータの一実施例を示す図である。

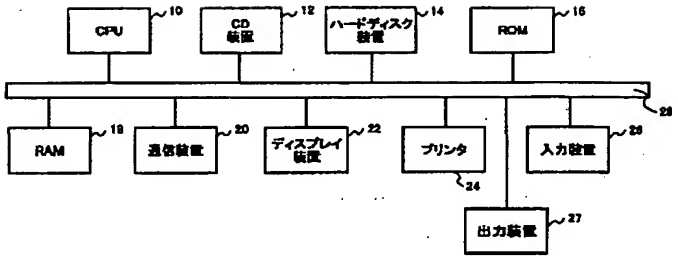
【図１６】生成されるプロファイルの一実施例を示す図である。

【図１７】ログデータの一実施例を示す図である。

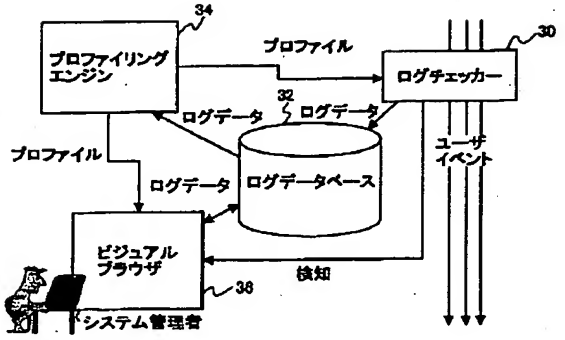
【符号の説明】

１０　ＣＰＵ
１２　ＣＤ装置
１４　ハードディスク装置
２０　通信装置
２２　ディスプレイ装置
２４　プリンタ
２６　入力装置
２７　出力装置
３０　ログチェッカー
３２　ログデータベース
３４　プロファイリングエンジン
３６　ビジュアルブラウザ
４０　ログ表示部
４３　Ｇｅｔボタン
４４　Ｃｌｅａｒボタン
４５　Ｂａｃｋボタン
４６　カッコーアイコン
５０，５２，５４，５６　指定欄
６０　楕円球
７０　プロファイル表示部

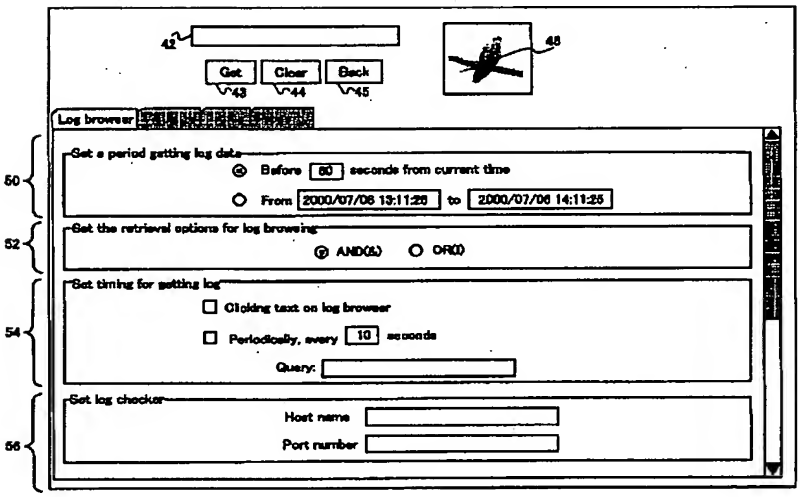
【図１】



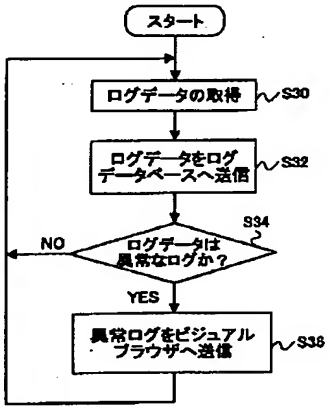
【図２】



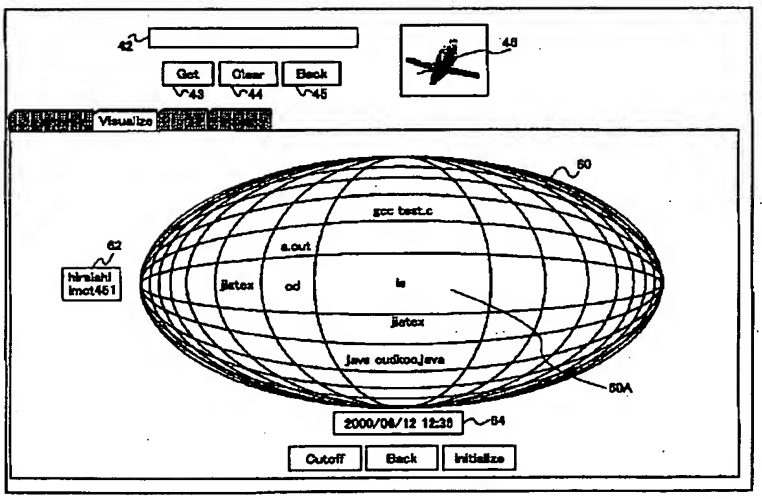
【図４】



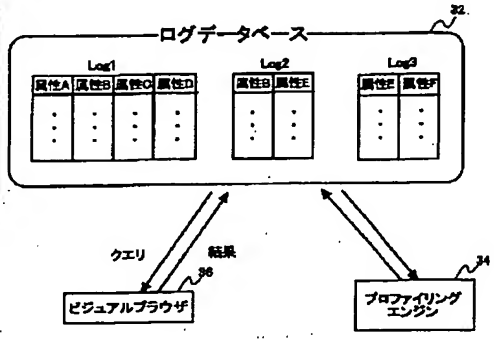
【図１０】



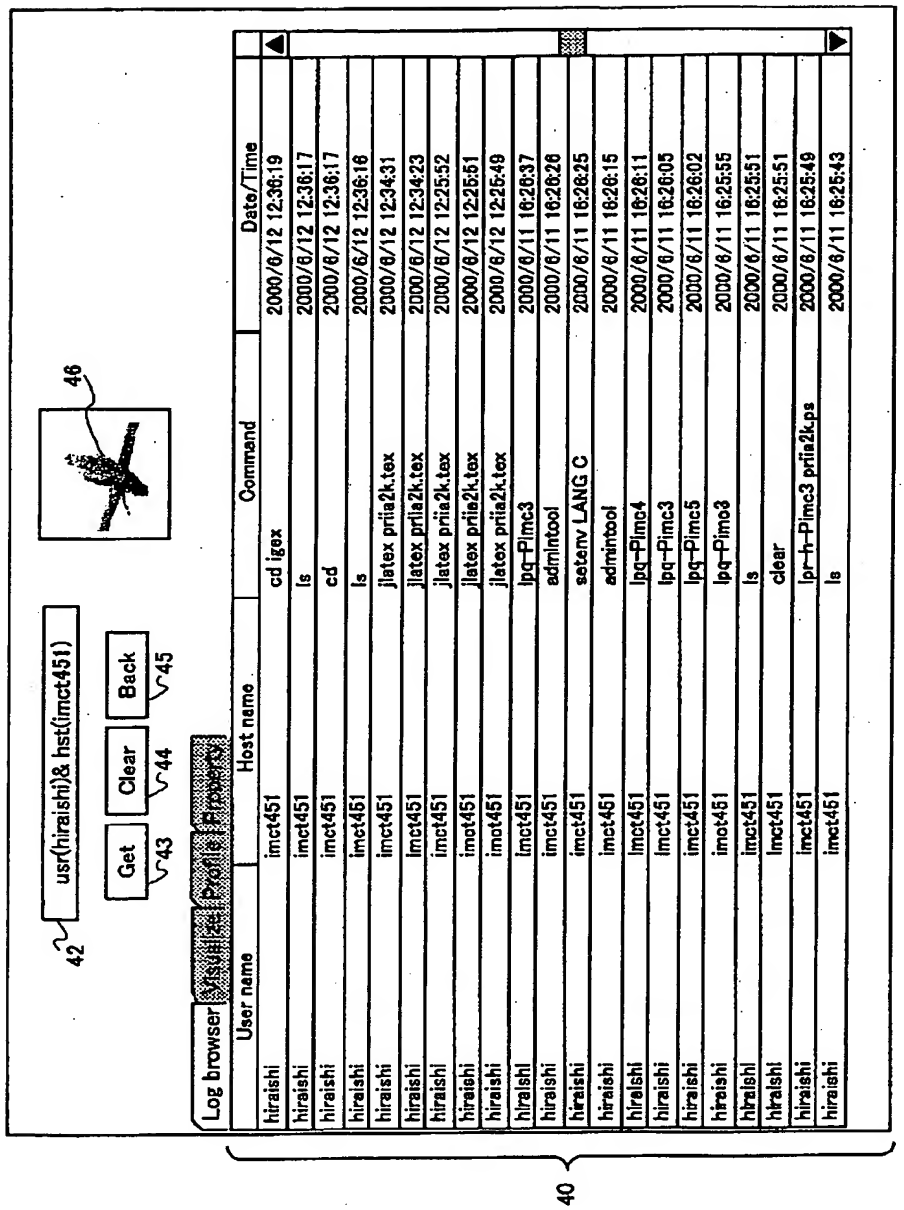
【図５】



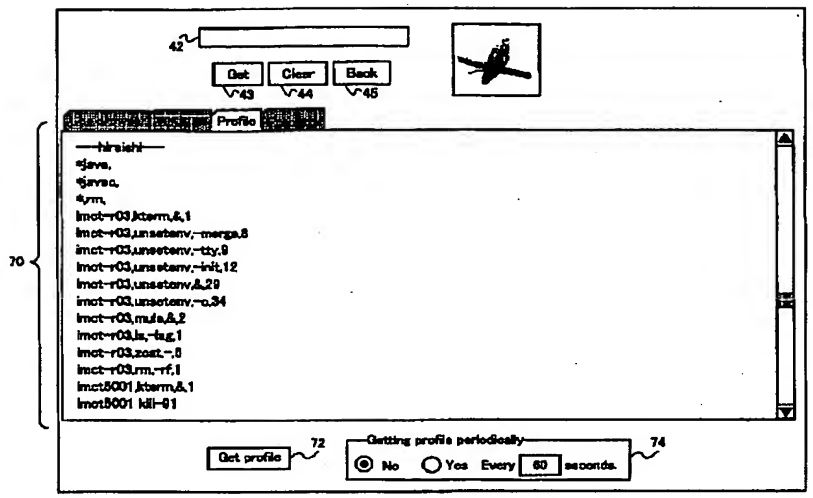
【図８】



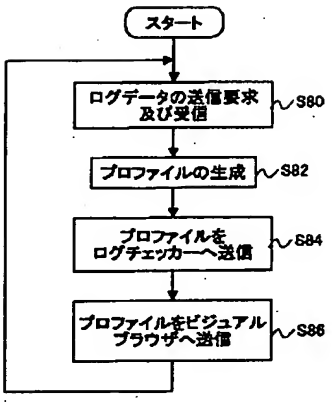
【図３】


40
Log browser
User Profile
Error
User name
Host name
Command
Date/Time
usr(hiraishi)& hst(imct451)
42
Get
43
Clear
44
Back
45
46
hiraishi imct451 cd igex 2000/6/12 12:36:19
hiraishi imct451 ls 2000/6/12 12:36:17
hiraishi imct451 cd 2000/6/12 12:36:17
hiraishi imct451 ls 2000/6/12 12:36:16
hiraishi imct451 jlatex pria2k.tex 2000/6/12 12:34:31
hiraishi imct451 jlatex pria2k.tex 2000/6/12 12:34:23
hiraishi imct451 jlatex pria2k.tex 2000/6/12 12:25:52
hiraishi imct451 jlatex pria2k.tex 2000/6/12 12:25:51
hiraishi imct451 jlatex pria2k.tex 2000/6/12 12:25:49
hiraishi imct451 lpq-Pimc3 2000/6/11 16:26:37
hiraishi imct451 admintool 2000/6/11 16:26:26
hiraishi imct451 setenv LANG C 2000/6/11 16:26:25
hiraishi imct451 admintool 2000/6/11 16:26:15
hiraishi imct451 lpq-Pimc4 2000/6/11 16:26:11
hiraishi imct451 lpq-Pimc3 2000/6/11 16:26:05
hiraishi imct451 lpq-Pimc5 2000/6/11 16:26:02
hiraishi imct451 lpq-Pimc3 2000/6/11 16:25:55
hiraishi imct451 ls 2000/6/11 16:25:51
hiraishi imct451 clear 2000/6/11 16:25:51
hiraishi imct451 lpr-h-Pimc3 pria2k.ps 2000/6/11 16:25:49
hiraishi imct451 ls 2000/6/11 16:25:43

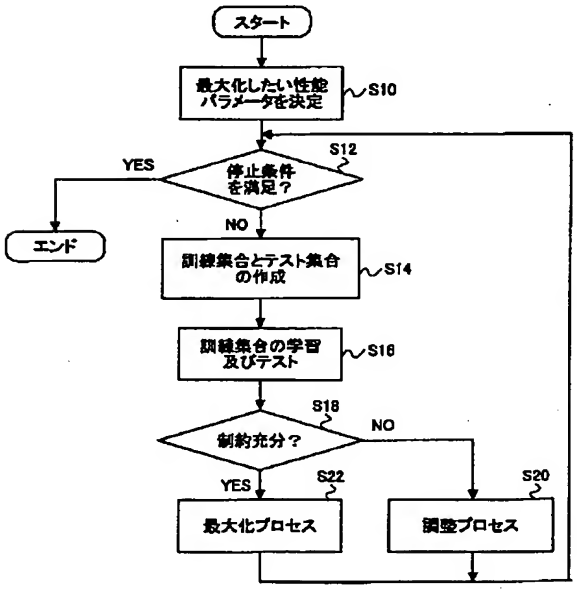
【図６】



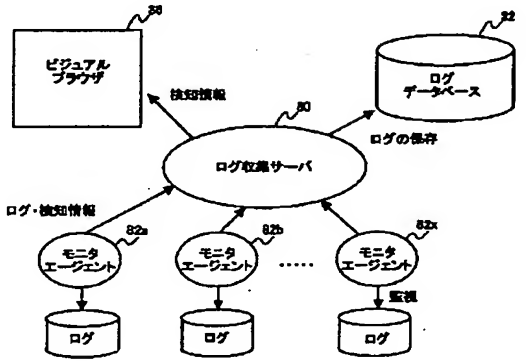
【図１４】



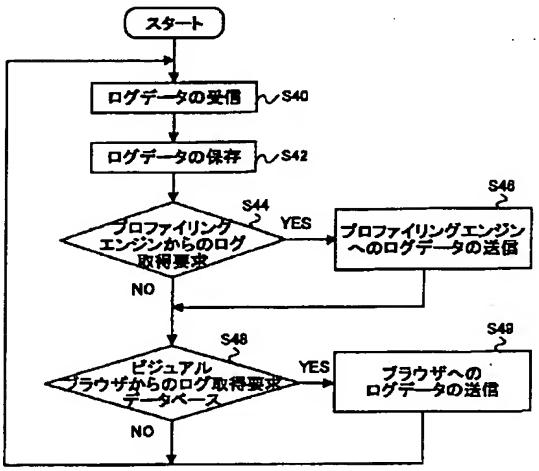
【図７】



【図９】



【図１１】



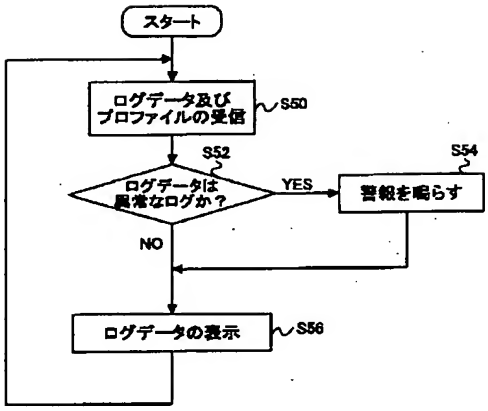
【図１５】

```
行
1  #+0984615658|imotut01
2  jlatex a.tex
3  #+0984615660|imotut01
4  dvi2ps a.dvi>a.ps
5  #+0984615676|imotut02
6  lpr-Pimc3 a.ps
7  #+0984615688|imotut02
8  jlatex b.tex
9  #+0984615690|imotut02
10 dvi2ps-t 1 b.dvi> b.ps
11 #+0984615706|imotut02
12 lpr-Pimc3 b.ps
```

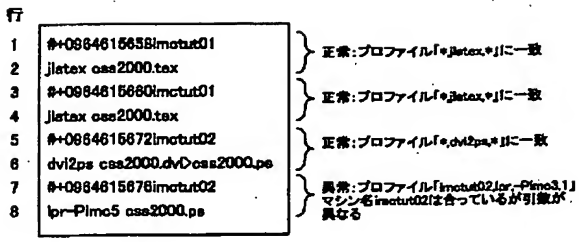
【図１６】

```
行
1  —yoshii—
2  *.jlatex,*
3  *.dvi2ps,*
4  imotut02,lpr,-Pimc3,1
```

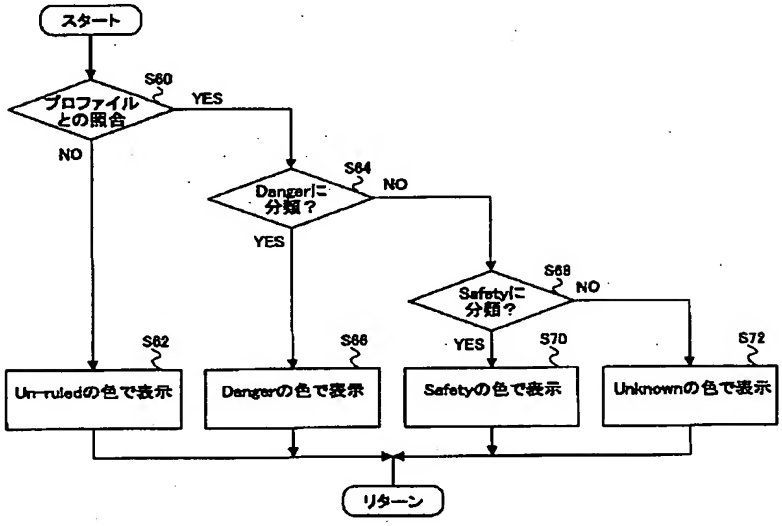
【図１２】


スタート
ログデータ及び
プロファイルの受信
S50
S52
ログデータは
異常なログか？
YES
S54
警報を鳴らす
NO
ログデータの表示
S56

【図１７】


行
1 #+0864615638imctut01
2 jlatex css2000.tex
3 #+0864615660imctut01
4 jlatex css2000.tex
5 #+0864615672imctut02
6 dvi2ps css2000.dvi>css2000.ps
7 #+0864615676imctut02
8 lpr-Plmc5 css2000.ps
正常：プロファイル「*jlatex.*」に一致
正常：プロファイル「*jlatex.*」に一致
正常：プロファイル「*.dvi2ps.*」に一致
異常：プロファイル「imctut02lpr.-Plmc3.1」
マシン名imctut02は合っているが引数が
異なる

【図１３】


スタート
プロファイル
との照合
S60
YES
NO
Dangerに
分類？
S64
NO
YES
Safetyに
分類？
S68
NO
YES
S62
Un-ruledの色で表示
S66
Dangerの色で表示
S70
Safetyの色で表示
S72
Unknownの色で表示
リターン